

# **Erstellung von Bibliographien auf der Basis von XML und XSL**

## **Diplomarbeit**

im Fach Datenbanksysteme und Aufbau von Datenbanken  
Studiengang Informationsmanagement  
der  
Fachhochschule Stuttgart –  
Hochschule der Medien

**Heike Müller**

|              |                              |
|--------------|------------------------------|
| Erstprüfer:  | Prof. Dr. Wolf-Fritz Riekert |
| Zweitprüfer: | Prof. Dr. Frank Thissen      |

Bearbeitungszeitraum: 01. August 2001 bis 05. November 2001

Stuttgart, November 2001

## Kurzfassung

Zugrunde liegendes Ziel der vorgestellten Arbeit ist, Literaturangaben durch technische Unterstützung in ein einheitliches Format zu bringen und deren Erstellung dadurch zu erleichtern. Dabei werden die Möglichkeiten der Extensible Markup Language (XML), der Extensible Stylesheet Language (XSL) und des Transformationsprogramms (XSLT) genutzt. Durch ein bereits existierendes Programm können Daten aus dem für Literaturdaten sehr verbreiteten Format BIBTEX in eine entsprechende XML-Darstellung gebracht werden. Aus dieser BIBTEXML-Darstellung von Literaturangaben sollen nun verschiedene Literaturlistenformate generiert werden. Dabei werden unterschiedliche Arten der Gestaltung und Reihenfolge von einzelnen Elementen einer Quellenangabe berücksichtigt. Eine Sortierfunktion wird ebenso ermöglicht wie eine Auswahl bestimmter Angaben aus einer großen Bibliographiedatei. Ziel ist zudem die Generierung eines Ausgabeformats, das in Winword übernommen werden kann. Praktische Anwendung finden die genannten Funktionalitäten z.B. in der Übernahme von Literaturangaben in eine Diplom- oder Masterarbeit.

**Schlagwörter:** XML, XSL, XSLT, Diplomarbeit, wissenschaftliche Arbeit, Bibliographie, Literaturangabe

## Abstract

The basic idea of this work is to provide technical support for the unified design of bibliographies. To achieve this goal a prototype has been developed by using the Extensible Markup Language (XML), Extensible Stylesheet Language (XSL) as well as XSL Transformations (XSLT). As a huge amount of bibliographic data is presented by using the BIBTEX format, a tool for the transformation of this data into XML representation has already been developed. Taking this BIBTEXML representation as a basis different style sheets have been designed in order to provide the possibility to unify bibliographies. This should be done by offering style sheets for various designs of bibliographic data, a function of sorting them and choosing a single element out of them. In addition, a format compatible with Winword should be generated. As an example a student should be given the possibility to insert such bibliographic data in his diploma thesis.

**Keywords:** XML, XSL, XSLT, thesis, bibliography

# Inhaltsverzeichnis

|   |           |
|---|-----------|
| <b>Kurzfassung .....</b>  | <b>2</b>  |
| <b>Abstract .....</b>   | <b>2</b>  |
| <b>Inhaltsverzeichnis.....</b>                                      | <b>3</b>  |
| <b>Abbildungsverzeichnis.....</b>                                   | <b>5</b>  |
| <b>Abkürzungsverzeichnis.....</b>                                   | <b>6</b>  |
| <b>1     Einleitung.....</b>  | <b>7</b>  |
| <b>2     Status.....</b>  | <b>9</b>  |
| 2.1   Das Format von Literaturangaben .....                         | 9         |
| 2.2   BIBTEX .....  | 9         |
| 2.3   BIBTEXML .....  | 11        |
| 2.4   Defizite.....   | 13        |
| 2.5   Ziele der Arbeit .....  | 13        |
| <b>3     Spezifikation der Aufgabenstellung .....</b>               | <b>15</b> |
| <b>4     XML und XSL.....</b>                                       | <b>16</b> |
| 4.1   XML.....  | 16        |
| 4.1.1 Die XML-Struktur .....  | 16        |
| 4.1.2 DTD und XML-Schema .....                                      | 17        |
| 4.1.3 Namensräume .....   | 17        |
| 4.1.4 XPath .....   | 17        |
| 4.2   XSL und XSLT .....  | 17        |
| <b>5     Generierung verschiedener Literaturlistenformate .....</b> | <b>19</b> |
| 5.1   Lösungsidee .....   | 19        |
| 5.1.1 Ansatz .....  | 19        |
| 5.1.2 Warum XML und XSL(T)? .....                                   | 19        |
| 5.2   Ausgangspunkt.....  | 20        |
| 5.2.1 Die BIBTEXML-Datei .....                                      | 20        |
| 5.2.2 Strukturvorgaben .....  | 21        |
| 5.3   Entwicklung eines Prototyps .....                             | 21        |
| 5.3.1 Ausgabeformat.....  | 21        |
| 5.3.2 XSLT-Stylesheets .....  | 25        |
| 5.3.3 Ergebnis .....  | 37        |

---

|          |  |           |
|----------|--|-----------|
| <b>6</b> | <b>Abschließende Zusammenfassung und Ausblick.....</b> | <b>39</b> |
|          | <b>Anhang A: Programmbeispiele.....</b>                | <b>41</b> |
| A.1      | Ausgabe eines Autors.....                              | 41        |
| A.2      | Ausgabe für Bücher und Beiträge aus Büchern .....      | 41        |
| A.3      | Ausgabe für Bücher und Beiträge aus Büchern .....      | 42        |
|          | <b>Anhang B: Elemente einer BIBTEXML-Datei.....</b>    | <b>44</b> |
|          | <b>Literaturverzeichnis.....</b>                       | <b>46</b> |
|          | <b>Erklärung.....</b>                                  | <b>47</b> |

## Abbildungsverzeichnis

|   |    |
|---|----|
| Abbildung 1: Umsetzung des generierten HTML-Codes im Browser.....   | 24 |
| Abbildung 2: An DIN 1505 angelehnte Form von Literaturangaben ..... | 33 |
| Abbildung 3: Alternative Darstellungsweise .....                    | 34 |
| Abbildung 4: Sortierung nach Verfasseramen .....                    | 35 |
| Abbildung 5: Einfügen einer HTML-Datei in Winword .....             | 37 |
| Abbildung 6: Online-Formular .....                                  | 39 |

## Abkürzungsverzeichnis

|      |                                |
|------|--------------------------------|
| CSS  | Cascading Style Sheets         |
| DTD  | Document Type Definition       |
| HdM  | Hochschule der Medien          |
| HTML | Hypertext Markup Language      |
| W3C  | World Wide Web Consortium      |
| XML  | Extensible Markup Language     |
| XSL  | Extensible Stylesheet Language |
| XSLT | XSL Transformations            |

## 1 Einleitung

Eine Bibliographie, wie sie üblicherweise einem veröffentlichten Dokument, etwa einer wissenschaftlichen Arbeit, angefügt ist, enthält bestimmte Angaben über die verwendete Literatur. Solche Angaben sind z.B. der Name des Verfassers einer Schrift, deren Titel und der Verlag, der die Schrift aufgelegt hat. Die für eine Titelangabe benötigten Daten sind wiederum abhängig von der Art der benutzten Quelle, sei es ein Buch, ein Zeitschriftenartikel oder auch eine WWW-Seite.

Beim Vergleich diverser Literaturverzeichnisse fällt jedoch auf, daß sich auch die Darstellungsformen der Quellenangaben unterscheiden. So gibt es etwa verschiedene Möglichkeiten der Schreibweise der einzelnen Angaben. Der Vorname eines Autors z.B. kann sowohl ausgeschrieben als auch abgekürzt werden, die Kursivschreibweise kann verwendet werden und auch die Angabe „Herausgeber“ kann entweder ausgeschrieben oder abgekürzt als „Hrsg.“ erscheinen.

Die Reihenfolge der einzelnen Elemente kann ebenso variieren wie die Deskriptionszeichen, die zur Trennung der Bestandteile verwendet werden. So können z.B. bei der Angabe von mehreren Autoren eines Beitrags deren Namen sowohl durch ein Komma als auch durch einen Semikolon voneinander getrennt werden.

Es wird schnell deutlich, daß es hinsichtlich der Vereinheitlichung von Quellenangaben innerhalb eines Werkes zu Schwierigkeiten kommen kann, besonders auch, wenn in der Titelangabe mehrere Autoren genannt werden sollen. Die Erzeugung verschiedener Bibliographiestile für diverse Einsatzmöglichkeiten stellt sich oft als sehr mühsam dar. So wird die traditionelle Erstellung einer Literaturliste von Hand oft zu einer zeit- und arbeitsintensiven Aufgabe.

Nun stehen online zahlreiche Bibliographien zur Verfügung, veröffentlicht im BIBTEX-Format, dem wohl meist verwendeten Format für Bibliographien. Aufgrund der Möglichkeit von BIBTEX, die Formvorgaben für bibliographische Angaben getrennt vom eigentlichen Inhalt abzuspeichern, können Daten leicht wiederverwendet werden. Für die Darstellung von Literaturangaben können also Stildateien angelegt werden, die eine Art Formatvorlage für die BIBTEX-Daten bilden. Dennoch weist BIBTEX einige Nachteile auf. So müssen die Daten in Ermangelung entsprechender Tools von Hand editiert werden, eine Tatsache, die zu einer uneinheitlichen Datenstruktur führt.

Diese Defizite von BIBTEX nahmen Angehörige der Eidgenössischen Technischen Hochschule (ETH) Zürich im Rahmen eines Projekts zum Anlaß, die Voraussetzungen für eine bessere Handhabung und dadurch eine effiziente Nutzung von BIBTEX-Bibliographien zu schaffen. Dafür realisierten sie ein System für die Darstellung von BIBTEX-Daten in einer XML-Umgebung, das sogenannte BIBTEXML.

Allerdings bleiben auch mit der Entwicklung von BIBTEXML noch einige Punkte offen, die im Hinblick auf die Unterstützung bei der praktischen Anwendung wie der Erstellung von Literaturverzeichnissen für wissenschaftliche Arbeiten von Nutzen wären. Da ja Literaturangaben, wie bereits vorher erwähnt, auf verschiedene Arten dargestellt werden können, wäre es sehr hilfreich, eine Unterstützung zur einheitlichen Erstellung eines Literaturverzeichnisses zur Verfügung zu haben. Dazu sollen Alternativen der Darstellung angeboten werden, wie eine Ausgabe von Literaturdaten angelehnt an die Norm DIN 1505<sup>1</sup> sowie weitere Darstellungsformen wie z.B. die Verwendung anderer Schriftstile und Deskriptionszeichen oder eine geänderte Reihenfolge der einzelnen Angaben. Zudem wäre es hilfreich, aus einer großen Bibliographiedatei einzelne Literaturangaben auswählen zu können, z.B. alle Werke, die von einem bestimmten Autor stammen. Da ein Großteil der wissenschaftlichen Arbeiten in Winword verfaßt wird, sollte außerdem ein entsprechendes Ausgabeformat generiert werden, um bibliographische Daten aus der BIBTEXML-Darstellung in Winword übernehmen zu können.

Für diese noch existierenden Defizite möchte ich nun durch den Einsatz der Möglichkeiten von XML bzw. XSL und XSLT eine denkbare Lösung erarbeiten.

---

<sup>1</sup> Deutsches Institut für Normung (DIN) (1984): DIN 1505, Teil 2. Titelangaben von Dokumenten: Zitierregeln. Beuth Verlag GmbH, Berlin.



## 2 Status

In diesem Kapitel stelle ich zunächst die Richtlinien zur Erstellung von Bibliographien kurz dar und skizziere dann die bisherigen technischen Entwicklungen auf diesem Gebiet.

### 2.1 Das Format von Literaturangaben

Bibliographische Quellenangaben sind nach DIN 1505, Teil 2 genormt, einer Norm, die Regeln für das Zitieren von Schrifttum und anderen Quellen definiert.

Sie dient der Identifizierung von Dokumenten und legt die erforderlichen Bestandteile einer Literaturangabe, ihre Reihenfolge und auch ihre Form fest.

Um einzelne Bestandteile eines Zitats voneinander abzuheben, werden sogenannte Deskriptionszeichen<sup>1</sup> verwendet.

DIN 1505 gibt Richtlinien hinsichtlich der Erstellung von Literatursammlungen, wie sie häufig am Ende Dokuments angehängt sind. Sie richtet sich dabei an sämtliche Verfasser solcher Literaturlisten.

Trotz der von dieser Norm festgelegten Regeln haben sich Abweichungen herausgebildet. So gibt es auf verschiedenen wissenschaftlichen Gebieten spezielle Richtlinien für die Erstellung von Bibliographien.

Üblicherweise sind Autoren einer Schrift oder eines Beitrags an die Vorgaben gebunden, die ihnen von ihren Verlegern auferlegt wurden.

Grundsätzlich kann gesagt werden, daß es trotz der fest definierten Regeln von DIN 1505 verschiedene Möglichkeiten gibt, Literaturverzeichnisse zu erstellen. Die Voraussetzung dabei ist allerdings, daß sie einheitlich sind.

### 2.2 BIBTEX

Erstellt man eine Literaturliste für ein Dokument auf traditionelle Art und Weise, d.h. ohne die gezielte Unterstützung eines Computers, sieht man sich schnell mit einigen Problemen konfrontiert.

Die Vereinheitlichung von Literaturangaben innerhalb eines Beitrags ist oft nicht einfach. Dies gilt besonders dann, wenn mehrere Autoren in einer Titelangabe zitiert werden sollen. Auch das Anlegen einer Datenbank von Literaturangaben, die für mehrere Beiträge zur Verfügung stehen sollen, ist keine leichte Aufgabe. Ein Literaturverzeich-

---

<sup>1</sup> Deskriptionszeichen sind durch die International Standard Bibliographic Description (ISBD) festgelegt

nis läßt sich nur mit großen Schwierigkeiten in eine andere Darstellungsform überführen. So fällt es etwa sehr schwer, eine alphabetisch erstellte Literaturliste in eine numerische Reihenfolge zu bringen.

Diese Schwierigkeiten machten die Notwendigkeit einer Unterstützung bei der Literaturverzeichniserstellung deutlich. Dabei wurde von den Möglichkeiten des von Leslie Lamport entwickelten, auf TEX basierenden Textsatzsystems LATEX ausgegangen. Als Seitenbeschreibungssprache bietet LATEX den Vorteil, durch Steueranweisungen ein professionelles Layout für Texte zu erzeugen, kompliziertere Textstrukturen wie z.B. Indizes, Verweise und Literaturverzeichnisse lassen sich auf einfache Art und Weise erzeugen. Einsatz findet LATEX daher vor allem in wissenschaftlichen Texten und hier besonders im Formelsatz.

BIBTEX ist ein von Oren Patashnik entwickeltes LATEX-Begleitprogramm, das als Hilfsmittel zur Erstellung von Literaturverzeichnissen unter LATEX dient. Es kann programmunabhängig eingesetzt werden. Ein Großteil der Bibliographien, die auch im Internet veröffentlicht sind, steht im BIBTEX-Format zur Verfügung.

Das Anlegen eines Literaturverzeichnisses mit BIBTEX läuft wie folgt ab: Die mit dem Befehl `\cite` gekennzeichneten Literaturverweise in einem LATEX-Dokument werden in eine Hilfsdatei `.aux` geschrieben. Diese wird dann von BIBTEX eingelesen. Anschließend werden die referenzierten Literaturverweise aus einer Referenzdatenbank, der `.bib`-Datei, eingelesen. Ausgegeben wird die sortierte Literaturliste als `.bbl`-Datei, in der die formatierten Literaturverweise enthalten sind. Deren Format ist dabei in der `.bst`-Stildatei definiert.

Es gibt eine Vielzahl solcher Stildateien, entwickelt von Einzelpersonen oder Organisationen, um bestimmten Formatvorgaben für Literaturangaben zu entsprechen, wie sie z.B. Verlage vorschreiben. Als Beispiele für BIBTEX-Stildateien können hier die Standard-BIBTEX-Stildatei `abbrv.bst` oder der BIBTEX-Stil für Astronomie `astron.bst` angeführt werden.

Ein BIBTEX-Eintrag besteht grundsätzlich aus dem Eintragstyp, z.B. „book“ oder „article“ mit einem vorangestellten `@`, einem Schlüsselwort, das die entsprechende Literatur identifiziert und auf den `\cite`-Befehl verweist sowie verschiedenen Feldern, bestehend aus dem Feldbezeichner, dem Gleichheitszeichen (=) und den entsprechenden Feldangaben wie z.B. „author“, „title“ oder „year“. Diese Angaben stehen entweder in geschweiften Klammern (`{ }`) oder in Anführungszeichen („“). Der gesamte BIBTEX-Eintrag ist in geschweiften (`{ }`) oder runden Klammern (`( )`) eingeschlossen.

Ein Großteil der Eintragstypen in BIBTEX sind Standard-Eintragstypen wie `@article`, `@book`, `@booklet`, `@conference`, `@inbook` usw., doch es können auch andere Eintragstypen wie z.B. `@patent` oder `@collection` auftreten.

Dazu gibt es Standard-Feldnamen wie „address“, „annote“, „author“, „booktitle“ usw. Auch hier werden zusätzlich andere Feldnamen wie z.B. „affiliation“ oder „abstract“ verwendet.

Bsp. für einen BIBTEX-Eintrag:

```
@Article{ieee-ec:Fogel:1997,  
  author = {David B. Fogel},  
  title = {Evolutionary Computation: A New Transactions},  
  journal = IEEE-EC,  
  year = 1997,  
  type = {Foreword},  
  volume = 1,  
  number = 1,  
  month = {April},  
  pages = {1-2}  
}
```

Im Internet finden sich zahlreiche BIBTEX-Bibliographien. Wie jedoch aus der vorangegangenen kurzen Beschreibung des BIBTEX-Formats ersichtlich, hat dieses Format durch die vielen verschiedenen Möglichkeiten der Literatureintragsgestaltung den entscheidenden Nachteil von Dateninkonsistenzen und uneinheitlichen Datenstrukturen. Diese Tatsache erschwert auch die Handhabung der Daten.

## 2.3 BIBTEXML

Realisiert wurde BIBTEXML im Rahmen eines Projekts während des Wintersemesters 2000/2001 an der Eidgenössischen Technischen Hochschule (ETH) Zürich. Eine entsprechende Diplomarbeit von Brenno Lurati und Luca Previtali<sup>1</sup> liegt vor. Das Ergebnis der Entwicklungen wird im Internet unter <http://bibtexml.org> präsentiert, wo auch die Funktionalitäten des Systems getestet und angewandt werden können.

Der Grundgedanke bei der Entwicklung von BIBTEXML ist eine Erstellung von Bibliographien mit XML, um mit Hilfe der Möglichkeiten von XML eine bessere Handhabbarkeit und Strukturierung bibliographischer Daten zu gewährleisten. XML kann problemlos editiert werden. Mit XSLT können XML-Daten dann in verschiedene Formate gebracht werden. Auf die Eigenschaften von XML sowie XSL und XSLT werde ich in Kapitel 4 detaillierter eingehen.

Da ein Großteil der Bibliographien im BIBTEX-Format vorliegen, sollen diese wiederverwendet werden. Es mußte also ein Weg gefunden werden, BIBTEX-Daten mit XML darzustellen. Das Resultat dieser Überlegungen ist BIBTEXML, die Darstellung von BIBTEX-Daten in einer XML-Umgebung.

---

<sup>1</sup> Lurati, B. und Previtali, L. (2000/2001): BIBTEXML. Eidgenössische Technische Hochschule, Zürich. Diplomarbeit.[http://bibtexml.org/documents/da\\_2001\\_02.pdf](http://bibtexml.org/documents/da_2001_02.pdf). Datum des Zugriffs: 04.11.2001

Zur Konvertierung der BIBTEX-Datei in eine XML-Datei wurde ein entsprechender XML-Parser implementiert.

Außerdem wurden Latex-spezifische Sonderzeichen, die in einigen BIBTEX-Einträgen verwendet werden, in eine XML-kompatible Darstellung gebracht.

Um die Struktur der BIBTEXML-Dateien festzulegen, werden zwei XML-Schemadateien eingesetzt: durch das erste XML-Schema werden die Standard-BIBTEX-Felder festgelegt, das zweite definiert alle zusätzlichen Felder. Für den Fall, daß XML-Schema nicht unterstützt wird, steht eine entsprechende Document Type Definition (DTD) zur Verfügung.

Es besteht außerdem die Möglichkeit, Daten aus dem XML-Format wieder zurückzuwandeln in die BIBTEX-Darstellung. Diese Konvertierung der XML-Datei in eine .bib-Datei geschieht durch den Einsatz eines XSLT Style-Sheets.

Nach der Umwandlung der BIBTEX-Daten in XML-Daten können diese in einer Online-Datenbank gespeichert werden.

Die im vorangegangenen Abschnitt als Beispiel aufgezeigte BIBTEX-Datei stellt sich nach der Konvertierung in eine XML-Darstellung wie folgt dar:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE bibliography SYSTEM "bibtexml.dtd">
<bibliography>
  <bibitem type="article" label="ieee-ec:Fogel:1997">
    <author>
      <firstname>David</firstname>
      <middlename>B.</middlename>
      <lastname>Fogel</lastname>
    </author>
    <title>Evolutionary Computation: { A } New Transactions</title>
    <journal>
      <abbrev alias="IEEE-EC"/>
    </journal>
    <year>1997</year>
    <type>Foreword</type>
    <volume>1</volume>
    <number>1</number>
    <month>
      <abbrev alias="apr"/>
    </month>
  </bibitem>
</bibliography>
```

```
</month>  
<pages>1--2</pages>  
</bibitem>  
</bibliography>
```

Auf die genauere Beschreibung des BIBTEXXML-Formats wird in Kapitel 5.2 eingegangen.

## 2.4 Defizite

Durch die Umwandlung von BIBTEX- in XML-Dateien mit Hilfe des in Abschnitt 2.3 erwähnten Konvertierungstools wird es nun also möglich, die bereits vorhandenen BIBTEX-Bibliographien zu nutzen, ihnen eine klare Struktur zu geben und somit leistungsfähige Bibliographie-Datenbanken zu entwickeln.

Dennoch läßt auch dieser Stand der Entwicklungen noch einige Verbesserungen zu.

Die bibliographischen Daten präsentieren sich nach ihrer Umwandlung aus dem BIBTEX-Format in der XML-typischen Baumstruktur und daher für einen konkreten Einsatz wie beispielsweise in einer wissenschaftlichen Arbeit oder auch im Internetauftritt einer Hochschule oder eines sonstigen Informationsangebotes in dieser Form eher ungeeignet. Eine entsprechende Formatierung für den Browser wäre daher sinnvoll.

Es wäre weiterhin von Nutzen, ein Literaturverzeichnis automatisch in eine einheitliche Form bringen zu können, dabei aber aus verschiedenen Alternativen auswählen zu können.

Oft wäre eine bessere Überschaubarkeit einer Bibliographie wünschenswert, gerade auch beim gezielten Interesse an bestimmten Fakten.

Besonders im Hinblick auf die praktische Anwendung, der Erstellung von Bibliographien für wissenschaftliche Arbeiten, die zumeist mit einem Textverarbeitungsprogramm - vorzugsweise mit Winword - geschrieben werden, wäre es außerdem sinnvoll, mit XML erstellte Literaturverangaben in Winword übernehmen zu können.

## 2.5 Ziele der Arbeit

Literaturdaten im XML-Format sollen nun auch konkret eingesetzt werden, sei es in Internetauftritten oder zur Unterstützung bei der Erstellung von den oft umfangreichen Literaturverzeichnissen wissenschaftlicher Arbeiten.

Ziel ist also zunächst, Literaturzitate so zu formatieren, daß sie eine einheitliche Form erhalten und auch aus der gewöhnungsbedürftigen XML-Baumstruktur in eine übliche Darstellung gebracht werden.

Literaturverzeichnisse sollten einheitlich erstellt werden, die einzelnen Angaben sollen, je nach Literaturquelle, immer die selben Bestandteile enthalten. Außerdem sollen De-

skriptionszeichen und Schriftstile einheitlich verwendet werden. Um dieses Defizit zu beheben, soll ein Weg geboten werden, eine Bibliographie automatisch in eine bestimmte Form zu bringen. Dabei soll zusätzlich die Möglichkeit bestehen, den gewünschten Stil für ein Literaturverzeichnis auszuwählen. Es soll eine an die normierte Darstellung von bibliographischen Angaben nach DIN 1505 angelehnte Form angeboten werden. Des weiteren wird eine alternative Formatierung entworfen, die sich hinsichtlich der Verwendung verschiedener Schriftstile und Deskriptionszeichen und auch einer geänderten Reihenfolge der einzelnen Angaben von der ersten Form unterscheidet.

Um eine bessere Übersichtlichkeit der Bibliographien zu gewährleisten, sollen diese sortiert werden können, beispielsweise nach Autor, Titel oder Erscheinungsjahr. Das Herausfiltern einzelner Angaben, z.B. des Erstautors, aus einer großen Bibliographie-datei, soll ermöglicht werden.

Um tatsächlich beim Verfassen einer wissenschaftlichen Arbeit zu unterstützen, sollen die XML-Dateien in ein Format gebracht werden, das problemlos in Winword übernommen werden kann.

### 3 Spezifikation der Aufgabenstellung

Zur Verdeutlichung der Anforderungen werde ich im folgenden ausführlicher auf Motivation und Ziele meiner Arbeit eingehen.

Grundgedanke ist, Literaturangaben für wissenschaftliche Arbeiten wie etwa Diplomarbeiten in ein einheitliches Format sowie in eine entsprechende Darstellung zu bringen und die Erstellung von Literaturverzeichnissen dadurch zu erleichtern.

Ausgehend von einer Repräsentation der Literaturangaben im XML-Format soll ein Prototyp für die Weiterverarbeitung der Daten entworfen werden, um die gewünschten Ziele hinsichtlich der Formatierung von Bibliographien zu erreichen.

Da ja an der ETH Zürich bereits Entwicklungen auf dem Gebiet der Literaturlistenerstellung mit Hilfe von XML bzw. XSL und XSLT gemacht wurden, war es sinnvoll, diese als Ausgangspunkt zu verwenden.

Die benötigten Testdaten für meine Arbeit beziehe ich aus den im Internet zur Verfügung stehenden BIBTEX-Bibliographien, die dann durch das Konvertierungsprogramm in ein XML-Format gebracht werden. Diese aus dem BIBTEX-Format konvertierten XML-Dateien, für die auch eine entsprechende DTD existiert, sind die Basis meiner Entwicklungen.

Der Einsatz von XML bzw. die anschließende Umwandlung und Formatierung mit Hilfe von XSL und XSLT soll das Verfassen von Bibliographien vereinfachen, deren Format vereinheitlichen, durch verschiedene Darstellungsweisen vielseitig einsetzbar und anschließend auch direkt in Textverarbeitungsprogrammen, insbesondere in Word, verwendbar gemacht werden.

Ziel meiner Arbeit ist jedoch nicht, Literaturverzeichnisse aus dem Internet zu übernehmen. Vielmehr soll ein Prototyp für eine komfortable Unterstützung beim Umgang mit Bibliographien geschaffen werden.

## 4 XML und XSL

Das folgende Kapitel befaßt sich mit der Extensible Markup Language (XML) und der Extensible Style Sheet Language (XSL). Dabei sollen die Möglichkeiten von XML und XSL aufgezeigt werden. Im Zusammenhang mit XSL werde ich näher auf XSL Transformations (XSLT), das Transformationsprogramm von XML, eingehen.

### 4.1 XML

Die Extensible Markup Language (XML) wurde vom World Wide Web Konsortium (W3C) als Meta-Auszeichnungssprache entwickelt die es erlaubt, eigenes Markup für Dokumente zu definieren. Im Gegensatz zur Hypertext Markup Language (HTML) arbeitet XML nicht mit fest vorgegebenen Befehlen („Tags“), vielmehr werden die Regeln und Befehle für ein in XML erstelltes Dokument von dessen Programmierer festgelegt. XML soll jedoch nicht der Nachfolger von HTML sein.

Ausschließlich die Struktur eines Dokuments wird durch XML festgelegt, die Darstellung im Browser muß durch Stylesheets bestimmt werden, in denen die Formatierungen für die einzelnen XML-Elemente definiert werden. Dazu können Cascading Style Sheets (CSS) oder auch die Extensible Stylesheet Language (XSL), auf die ich in Kapitel 4.2. näher eingehen werde, verwendet werden.

#### 4.1.1 Die XML-Struktur

XML-Dokumente setzen sich aus reinen Textzeichen zusammen und gründen auf dem Unicode-Zeichensatz. Der Text eines XML-Dokuments besteht aus XML-Auszeichnungen und Zeichendaten.

Zur Identifikation eines Dokuments als XML-Dokument durch den Prozessor muß die XML-Datei mit einer XML-Deklaration, innerhalb dieser Angaben zur näheren Beschreibung des XML-Dokuments gemacht werden können, versehen werden. Hierbei erfolgt zunächst eine Spezifizierung der verwendeten XML-Version, bei der es sich zum Zeitpunkt des Verfassens dieser Arbeit um die Version 1.0 handelt, sowie die Angabe des benutzten Zeichensatzes.

Ein XML-Dokument besteht aus verschiedenen, ineinander verschachtelten Elementen, umschlossen vom sogenannten Wurzelement. So entsteht folgende Struktur:

```
<Wurzelement>
  <Element1></Element1>
  <Element2></Element2>
</Wurzelement>
```



Um vom XML-Prozessor, dem Parser, angezeigt werden zu können, muß ein XML-Dokument wohlgeformt sein, d.h. die Syntax muß korrekt und bestimmte Regeln erfüllt sein.

### 4.1.2 DTD und XML-Schema

Die Struktur eines XML-Dokuments kann durch eine Document Type Definition (DTD) definiert werden. Hier werden die Regeln für Dokumente festgelegt, sämtliche Elemente müssen in der DTD eingetragen werden. Bestimmungen für Dokumente mit einer identischen Struktur können in einer einzigen DTD definiert werden.

Einen neuen Ansatz, Regeln für ein XML-Dokument ohne die Verwendung einer DTD zu bestimmen, bietet XML-Schema. Im Gegensatz zur DTD verwendet XML-Schema XML-Syntax, unterstützt dazu viele verschiedene Datentypen und erlaubt den Einsatz von Namensräumen.

### 4.1.3 Namensräume

Namensräume sind eine Spezifikation hinsichtlich der Namen für Elementtypen und Attribute, die es ermöglicht, diese Elementtypen und Attribute eindeutig zu identifizieren. So kann Mehrdeutigkeit von Information vermieden werden.

### 4.1.4 XPath

Ein weiterer auf XML basierender Standard ist XPath. Dies ist eine Sprache, die die Möglichkeit bietet, Teile eines XML-Dokuments gezielt zu adressieren, Zeichenketten zu manipulieren oder Muster für Übereinstimmungen mit Knoten der XML-Datei zu erstellen. Dabei geht XPath von einer baumartigen Struktur eines XML-Dokuments aus.

## 4.2 XSL und XSLT

Die Extensible Stylesheet Language (XSL) beinhaltet zwei Stylesheet-Sprachen, die unabhängig voneinander eingesetzt werden können: die Transformationssprache XSLT und die Formatierungssprache XSL-FO, auf die ich im Rahmen meiner Arbeit nicht näher eingehe.

Ein XML-Dokument kann mit Hilfe einer Transformation aufgrund der im XSL-Stylesheet definierten Regeln als neues XML-Dokument zurückgegeben oder in andere Formate wie etwa HTML, Text oder PDF konvertiert werden. Dabei verarbeitet der Prozessor sowohl das XML-Dokument als auch das XSLT-Stylesheet. Für die Festlegung der Transformationsregeln wird XML verwendet, d.h. Transformations-Stylesheets sind wohlgeformte XML.

Die durch eine Transformation erzeugten Dokumente können sich entweder auf die DTD des Quelldokuments beziehen oder aber eine neue DTD zugewiesen bekommen.

Die Verarbeitung eines Dokuments wird durch Schablonen („templates“) bewirkt. Sie legen fest, welche Elemente des Quelldokuments ausgegeben werden sollen und in welcher Form diese Ausgabe erfolgen soll. Durch Muster („patterns“) werden die Knoten im Ausgangsdokument bestimmt, auf die sich eine Schablone bezieht.

XSLT ist Transformationssprache und Bestandteil von XSL. XSLT-Dokumente setzen sich aus einer Ansammlung von Regeln zusammen, die aus einer Schablone und einem Muster bestehen und während des Umwandlungsprozesses nacheinander aufgerufen werden.

## 5 Generierung verschiedener Literaturlistenformate

In diesem Kapitel wird nun ausführlich auf meine Arbeit eingegangen, den von mir vorgeschlagenen Lösungsweg werde ich detailliert vorstellen und erläutern.

Zunächst wird die Struktur der BIBTEXML-Datei, der Quelldateien für die spätere Umwandlung mit XSLT, beschrieben.

Danach werde ich darlegen, wie eine derartige XML-Datei umgewandelt werden kann, um damit verschiedene Bibliographie-Stile zu erhalten.

Zum besseren Verständnis sind hier die Namen von XML-Elementen - entsprechend der tatsächlichen Darstellungsweise in Dokumenten - in spitze Klammern (<>) eingefaßt, XML-Attribute habe ich *kursiv* geschrieben. BIBTEX-Eintragstypen haben, gemäß einer originalen BIBTEX-Literaturangabe, ein vorangestelltes @.

Von mir generierten oder sonstigen relevanten Programmcode habe ich zur Verdeutlichung blau eingefärbt.

Die XSLT-Stylesheets habe ich im Editor XML Spy programmiert, der auch einen eingebauten Parser besitzt.

### 5.1 Lösungsidee

#### 5.1.1 Ansatz

Anhand des vorgestellten Stands der Dinge wird schnell klar, daß hinsichtlich des Umgangs mit Literaturdaten noch einige Verbesserungen notwendig sind.

Zwar wurde mit BIBTEXML eine Möglichkeit geschaffen, Bibliographiedaten aus der BIBTEX-Darstellung in ein geeignetes XML-Format zu bringen und ihnen durch die Definition einer DTD eine feste Struktur zu geben, jedoch fehlt bis jetzt die geeignete Präsentation dieser Daten in einem einheitlichen Format und dadurch die Unterstützung beim aktiven Verfassen eines Literaturverzeichnis.

Hierfür möchte ich eine Lösung vorschlagen, die auf die durch die DTD der XML-Dateien fest vorgegebene Struktur der Daten aufsetzt. Meine Idee ist, die Literaturdaten im XML-Format mit den Möglichkeiten von XSLT in andere Stile und Formate umzuwandeln.

#### 5.1.2 Warum XML und XSL(T)?

Mit XML lassen sich komplexe Dokumentstrukturen klar und übersichtlich darstellen, Daten können leicht bearbeitet werden.

Mit den Möglichkeiten des Transformationsprogramms XSLT lassen sich XML-Dateien in verschiedene Formate wie z.B. HTML, Text oder PDF überführen. So können als XML-Dateien vorliegende Literaturdaten in ein Format gebracht werden, das in Textverarbeitungsprogramme wie Winword eingefügt werden kann, so daß es ermöglicht wird, mit XML erstellte Bibliographien direkt in einer wissenschaftlichen Arbeit zu verwenden.

Die Daten können außerdem durch verschiedene Stylesheets in unterschiedliche Darstellungsformen gebracht werden. Große Bibliographiedateien können mit Hilfe von XSLT-Anweisungen nach diversen Kriterien sortiert, einzelne Angaben können ausgewählt werden.

Die vorliegenden XML-Daten können in ein HTML-Format gebracht und so sowohl in Winword als auch in Internetauftritten etwa von Hochschulen verwendet werden.

## 5.2 Ausgangspunkt

### 5.2.1 Die BIBTEXML-Datei

Die generierte XML-Datei ist nahezu identisch strukturiert wie ihre Quelldatei im BIBTEX-Format. Aus diesem Grund enthält sie auch einige Elemente und Attribute, die aus speziellen BIBTEX-Angaben resultieren.

Um die Struktur des BIBTEXML-Formats zu verdeutlichen, sei hier ein weiteres Beispiel für eine derartige XML-Datei angeführt:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE bibliography SYSTEM "bibtexml.dtd">
<bibliography>
  <bibitem type="book" label="PRNN:Ripley:1996">
    <author>
      <firstname>B.</firstname>
      <middlename>D.</middlename>
      <lastname>Ripley</lastname>
    </author>
    <title>Pattern Recognition and Neural Networks</title>
    <publisher>Cambridge University Press</publisher>
    <address>Cambridge</address>
    <year>1996</year>
    <ISBN>0-521-46086-7</ISBN>
    <notstandard name="comment">comment</notstandard>
    <abbrev alias="PRNNref"/>
  </bibitem>
```

</bibliography>

Das Wurzelement einer BIBTEXML-Datei ist <bibliography>, das optional die das Dokument näher beschreibende Attribute *version*, *author*, *created* oder *lastmodified* enthält. Die Unterelemente von <bibliography> sind <macro>, <preamble>, <comment> und <bibitem>. Das Element <bibitem> deckt sämtliche BIBTEX-Eintragstypen wie beispielsweise @article oder @book ab. Der Typ der Literaturquelle wird im XML-Dokument durch eines der beiden im Element <bibitem> enthaltenen Attribute, dem Attribut *type*, festgelegt. Das zweite Attribut ist *label* und erfüllt die Aufgabe einer ID von <bibitem>.

Desweiteren enthält <bibitem> Unterelemente, die im Anhang einzeln aufgelistet sind. Sie entsprechen den Feldbezeichnern in BIBTEX. Dabei kann das XML-Element <author>, einem Unterelement von <bibitem>, die Unterelemente <firstname>, <middle-name>, <lastname> und <suffix> beinhalten.

Außerdem wurden die zusätzlichen Elemente <tex> und <abbrev> definiert, die eine volle Anlehnung an das BIBTEX-Format gewährleisten sollen. Durch das Element <abbrev> kann dem Makro-Mechanismus aus BIBTEX entsprochen werden, <tex> wird, vereinfacht gesagt, in Bezug auf spezielle Symbole und Zeichen verwendet.

### 5.2.2 Strukturvorgaben

Die Struktur dieser XML-Dateien ist durch XML-Schema bzw. alternativ dazu durch eine DTD festgelegt für den Fall, daß XML-Schema nicht unterstützt wird. Es werden zwei Schemadateien verwendet: eine für standardmäßig in BIBTEX auftretende Felder sowie eine für Felder, die vom Standard abweichen.

## 5.3 Entwicklung eines Prototyps

Der von mir entwickelte Prototyp besteht aus verschiedenen XSLT-Stylesheets, die die XML-Dateien dahingehend formatieren bzw. umwandeln, mein intendiertes Ziel zu erreichen.

In einem weiteren Schritt könnte noch eine Formularschnittstelle realisiert werden, die es möglich macht, eine bestimmte XML-Datei aus dem Dateisystem auszuwählen und ihr ein Stylesheet zuzuweisen, das diese Datei in das gewünschte Ausgabeformat bringt.

### 5.3.1 Ausgabeformat

Nach reiflicher Überlegung habe ich mich dazu entschlossen, die zu transformierenden XML-Dateien im HTML-Format ausgeben zu lassen.

HTML-Dateien können im Browser problemlos angezeigt werden, die etwas ungewohnte Baumanzeige einer reinen XML-Datei wird ersetzt durch formatierten Text. Bibliographien können so leicht in Webauftritte eingefügt werden. Außerdem ist die

HTML-Ausgabe interessant im Bezug auf den späteren Einsatz in Winword. In Kapitel 5.3.2.8 wird dieses Thema detailliert behandelt.

### 5.3.1.1 Spezifikation des Ausgabeformats

Festgelegt werden kann das Ausgabeformat durch das XSL-Element `<xsl:output>`. Im Fall von HTML bedeutet das folgende Angabe:

```
<xsl:output method="html"/>
```

Die vollständige Spezifikation der Ausgabe, wie ich sie in meinen Stylesheets verwenden, ist die nachfolgend angegebene:

```
<xsl:output method="html" encoding="ISO8859-1" doctype-public="-\\W3C\\DTD HTML 4.0 Transitional\\EN" indent="yes"/>
```

```
<xsl:strip-space elements="*" />?
```

Durch das Attribut *encoding* kann der im HTML zu verwendende Zeichensatz spezifiziert werden – im Falle der von mir entwickelten Stylesheets ist das ISO8859-1. ISO8859-1 deshalb, weil dadurch die deutschen Umlaute abgedeckt werden.

Ist für das HTML-Dokument ein `<head>` -Element vorgesehen, wird direkt nach dem Start-Tag dieses Elements ein `<meta>` -Element ausgegeben, das die Zeichencodierung bestimmt. Nachfolgendes Beispiel zeigt das `<meta>` -Element, das aufgrund der oben angeführten Spezifikation ausgegeben wird:

```
<head>
```

```
<META http-equiv="Content-Type" content="text/html; charset=UTF-16">
```

```
<title>Bibliography</title>
```

```
</head>
```

Eine DTD kann durch das Attribut *doctype* definiert werden – in den von mir entworfenen Stylesheets wird der öffentliche Bezeichner *doctype-public* verwendet. Dies bewirkt, daß das erste Element des HTML-Codes durch eine DTD der folgenden Art angeführt wird:

```
<!DOCTYPE html PUBLIC "-\\W3C\\DTD HTML 4.0 Transitional\\EN">
```

Das Attribut *indent* legt fest, ob das Output-Dokument automatisch eingerückt werden soll.

Die Default-Version des HTML-Outputs ist 4.0, d.h. das ausgegebene HTML-Dokument sollte den Vorgaben der HTML 4.0 Recommendation des W3C entsprechen.

Für leere Elemente soll in HTML kein End-Tag ausgegeben werden. Das bedeutet, daß Angaben im Stylesheet wie

```
<br/> oder <br></br>
```

```
als
```

```
<br>
```

ausgegeben werden soll und nicht in der Form

<br></br>.

### 5.3.1.2 Definition der Gestaltung des Outputs

Der gesamte Output soll zunächst auf weißem Hintergrund erfolgen, weshalb ich auch keine spezielle Hintergrundfarbe angegeben habe. Dafür habe ich mich besonders auch im Hinblick auf den späteren Einsatz der Literaturdaten in wissenschaftlichen Arbeiten entschieden.

Für eine Verwendung in Internet-Seiten kann der Hintergrund jederzeit problemlos angepaßt werden.

Die Ausgabe erfolgt in unsichtbaren Tabellen, d.h. für jede einzelne Literaturangabe einer großen Bibliographiedatei wird eine Tabelle erzeugt. Auch diese Formatierung habe ich besonders hinsichtlich der gewünschten Verwendung in Winword gewählt. In Kapitel 5.3.2.8 werde ich diesen Umstand näher beschreiben. In Bezug auf das zugrunde liegende XML-Dokument bedeutet diese Darstellungsform, daß jedes <bibitem> - Element in eine separate Tabelle geschrieben wird.

Ein XML-Dokument der Form

```
<?xml version="1.0" encoding="ISO8859-1"?>
<bibliography>
<bibitem type="book" label="PRNN:Abramowitz+Stegun:1965">
  <author>
    <firstname>M.</firstname>
    <lastname>Abramowitz</lastname>
  </author>
  <author>
    <firstname>I.</firstname>
    <middlename>A.</middlename>
    <lastname>Stegun</lastname>
  </author>
  <title>Handbook of Mathematical Functions with Formulas, Graphs and Mathematical
Tables</title>
  <publisher>Dover</publisher>
  <address>New York</address>
  <year>1965</year>
  <notstandard name="comment">comment</notstandard>
  <abbrev alias="PRNNref"/>
</bibitem>
</bibliography>
```

wird als folgenden HTML-Code ausgegeben:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
<META http-equiv="Content-Type" content="text/html; charset=UTF-16">
<title>Bibliography</title>
</head>
<body>
<font face="Times New Roman" size="3">
<table border="1">
<tr>
<td>
<font style="font-variant:small-caps">Abramowitz</font>
<text>,</text>M<text>.</text><text>;</text>
<font style="font-variant:small-caps">Stegun</font>
<text>,</text>I<text>.</text><text>,</text>A.<text>:</text>
<i>Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables</i>
<text>.</text>New York<text>:</text>Dover<text>,</text>1965
</td>
</tr>
</table>
</font>
</body>
</html>
```

Wie dieser HTML-Code im Browser umgesetzt wird, zeigt die folgende Abbildung. Zu Demonstrationszwecken habe ich für dieses Beispiel dem Rahmen der Stärke 1 mitgegeben.

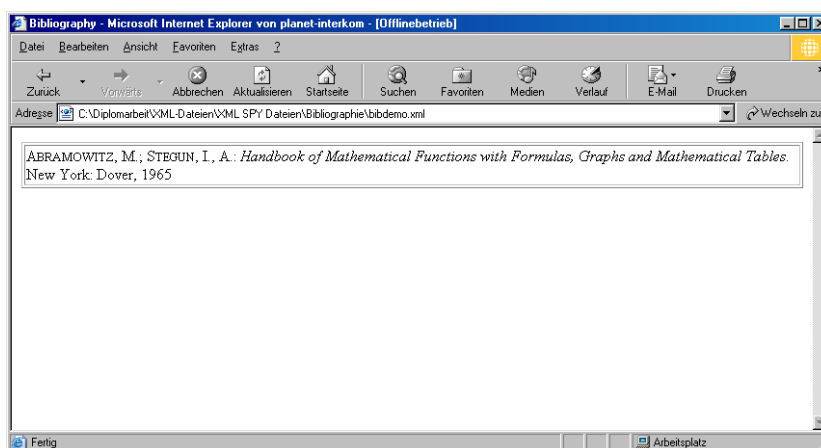


Abbildung 1: Umsetzung des generierten HTML-Codes im Browser



Als Schriftart für HTML-Dokument habe ich Times New Roman gewählt, da diese Schriftart für viele wissenschaftliche Arbeiten verwendet wird. Problemlos kann aber statt dessen auch die Schriftart Arial eingesetzt werden. Die Schriftgröße ist auf 3 festgelegt.

Im Stylesheet wird Definition der Schrifttypisierung durch folgenden HTML-Befehl gemacht, der dann auch im HTML-Dokument angegeben wird:

```
<font face="Times New Roman" size="3">
```

### 5.3.2 XSLT-Stylesheets

In einer Bibliographie muß sämtliche verwendete Literatur verzeichnet werden. Dabei ist es erforderlich, den Medientyp, aus dem die Literatur bezogen wurde, zu berücksichtigen. Bücher müssen in einem Literaturverzeichnis anders dargestellt werden als beispielsweise Zeitungsartikel. Je nach Literaturtyp variieren die benötigten Angaben, die Reihenfolge der Angaben und die Deskriptionszeichen.

Mein erster Schritt war also, für jeden Literaturtyp eine entsprechende Vorlage zu schaffen. Ich habe mich dazu auf die Quellen Buch, Beitrag aus einem Buch, Zeitschriftenaufsatz und WWW-Seite konzentriert. Die Vorlage soll zunächst lediglich dazu dienen, die pro Literaturtyp benötigten Angaben aus einem XML-Dokument herauszufiltern, verschiedene Arten der Darstellung werden in Kapitel 5.3.2 behandelt. Auch die Reihenfolge der Ausgabe bleibt noch unberücksichtigt.

Zunächst ist es erforderlich, den Literaturtyp in der XML-Datei identifizieren zu können. Diese Möglichkeit besteht, und zwar durch das im Element `<bibitem>` verwendete Attribut *type*. Ein Eintrag dieser Art sieht folgendermaßen aus:

```
<bibitem type="book" label="PRNN:Abramowitz+Stegun:1965">
```

Der Wert von *type* gibt also den Literaturtyp an, in diesem Fall *book*. *label* hat die Funktion einer ID für jedes `<bibitem>`-Element.

Den Zugriff auf den Wert des Attributs *type* erlaubt der XPath-Ausdruck `@type`. In meinem Stylesheet verwende ich daher, um beim oben angeführten Beispiel zu bleiben, für den Ausgabotyp Buch folgende Anweisung:

```
<xsl:if test="@type[.='book']">...</xsl:if >
```

Konkret bedeutet das: wenn der Literaturtyp ein Buch ist, sollen die Ausgabeanweisungen befolgt werden, die für ein Buch gelten. Diese werden dann zwischen Start- und End-Tag der `xsl:if`-Anweisung eingefügt. Dieser Teil wird nachfolgend skizziert.

#### 5.3.2.1 Bücher als selbständige Quelle

Bücher, die als Komplettwerk im Literaturverzeichnis angegeben werden, bezeichnet man als selbständige Quelle. Die Angaben Verfasser, Herausgeber, Institution oder Organisation sind für alle von mir behandelten Literaturquellen identisch und werden nachfolgend als Urheber bezeichnet, um meine Erläuterungen übersichtlicher zu gestalten.

ten. Urheber stehen am Anfang einer Literaturangabe. Entsprechend sind auch meine Stylesheets aufgebaut: zuerst wird zentral die Ausgabe für den Urheber definiert, dann folgt im Fall einer an DIN 1505 angelehnten Form der Sachtitel und anschließend wird der entsprechende Literaturtyp referenziert, um die spezifische Gestalt festzulegen. Exemplarisch erläutere ich im folgenden alle benötigten Anweisungen für Bücher, in den weiteren Kapiteln gehe ich dann auf die Besonderheiten anderer Quellentypen ein.

Literaturangaben von Büchern müssen grundsätzlich folgende Angaben enthalten:

- Verfasser
- oder
- Herausgeber mit nachgestelltem Hinweis „Hrsg.“
  - Sachtitel
  - Ausgabebezeichnung
  - Erscheinungsort
  - Verlag
  - Erscheinungsjahr

Der Verfasser kann dabei folgende Angaben umfassen:

- Nachname
- Erster Vorname
- Zweiter Vorname

Im XML-Dokument entspricht dies den folgenden Elementen:

- <author>
- oder
- <editor>
- oder
- <institution>
- oder
- <organization>
  - <title>
  - <edition>
  - <address>
  - <publisher>
  - <year>

Das Element <author> kann folgende Unterelemente beinhalten:

- <lastname>
- <firstname>

- <middlename>
- <suffix>

Erläuterungen zu den oben angeführten Elementen:

- <suffix> wird für Zusätze zum Namen wie Jr., III. usw. gebraucht und findet wohl vor allem im angelsächsischen Sprachraum Verwendung.
- <author> entspricht dem Verfasser, <editor> dem Herausgeber. Als Herausgeber können auch Institutionen oder Organisationen auftreten.

Die Werte der oben aufgeführten XML-Elemente müssen nun aus dem XML-Dokument herausgefiltert werden. Deskriptionszeichen sind im XSLT-Dokument in ein <text> - Element eingefaßt:

<xsl:text>, </xsl:text>

Das Problem war nun, daß diese Zeichen zur Trennung der einzelnen Angaben voneinander wie Komma, Semikolon, Punkt oder Doppelpunkt ausgegeben wurden, auch wenn keine Angaben existierten, die sie trennen sollten. So mußte ich die Ausgabe eines gesamten optionalen Elements an eine Bedingung knüpfen, in die ich auch das nachfolgende Deskriptionszeichen mit einschloß.

Ein Beispiel soll dies verdeutlichen:

```
<xsl:if test="count(edition)>0">
    <xsl:value-of select="edition"/>
    <text>.</text>.
</xsl:if>
```

Der korrekte Output des Verfassers stellte sich als der umfassendste Teil des Programms heraus, nicht zuletzt wegen seiner vier Unterelemente.

Grundsätzlich gilt:

- Bei Verfasserangaben werden Adelstitel (z.B. Baron) und Titulaturen (z.B. Dr.) weggelassen.
- Zusätze zum Nachnamen wie „von“ werden im XML-Dokument als Teil des Nachnamens, also <lastname>, angesehen

Der Output erfolgt immer in der Reihenfolge

- <lastname>
- <suffix>
- <firstname>
- <middlename>.

Gemäß DIN 1505-1 werden sowohl Nachname als auch erster Vorname in voller Länge wiedergegeben, ein zweiter Vorname wird abgekürzt.

Wie schon oben erwähnt, gehe ich an dieser Stelle nicht näher auf die Darstellung ein. Ich möchte hier lediglich anhand eines Beispiels demonstrieren, wie Verfasserangaben aus einem BIBTEXML-Dokument mit Hilfe von XSLT in eine entsprechende HTML-Form gebracht werden können. Hinsichtlich anderer, von DIN 1505 abweichender Darstellungsformen sei auf Kapitel 5.3.2.5 verwiesen.

Zusätzlich muß die Anzahl der Verfasser im zugrunde liegenden XML-Dokument berücksichtigt werden. Diese Anforderung stellt sich ebenso aufgrund der Deskriptorenzeichen.

Auch hier zeige ich zum besseren Verständnis den Fall der DIN 1505-konformen Wiedergabe auf.

- Ist ein Verfasser angegeben, soll dem letzten Namensteil ein Doppelpunkt folgen.
- Sind zwei Verfasser angegeben, sollen die beiden Verfasseramen durch Semikolon voneinander getrennt werden.

Dem letzten Namensteil des zweiten Verfasseramens soll ein Doppelpunkt folgen.

- Sind drei Verfasser angegeben, sollen die drei Verfasseramen durch Semikolon voneinander getrennt werden.

Dem letzten Namensteil des zweiten Verfasseramens soll ein Doppelpunkt (:) folgen.

- Bei mehr als drei Verfassern wird nur der erste Verfasseramen angegeben, die übrigen werden durch ein et al. ersetzt.

Der letzte Punkt ist nicht DIN-konform, da DIN vorschreibt, bei mehreren Verfassern alle anzugeben. Ich habe mich dennoch für diese Form der Wiedergabe entschieden da ich denke, daß sie übersichtlicher und absolut ausreichend ist.

Wegen des großen Umfangs des kompletten Befehlssatzes zur Verfasserausgabe habe ich im Anhang den Programmcode für die Ausgabe eines Verfassers aufgeführt.

Wenn im XML-Quelldokument kein Verfasser der Form <author> angegeben ist, können die Elemente <publisher>, <institution> oder <organization> als Verfasser fungieren. Der entsprechende Quelltext erscheint in der folgenden Form, allerdings zur besseren Überschaubarkeit ohne Textgestaltungs-elemente:

```
<xsl:if test="count(author)=0">
  <xsl:if test="count(editor)>0">
    <xsl:value-of select="editor"/>
    <text>(Hrsg.), </text>
  </xsl:if>
</xsl:if>
<xsl:if test="count(author)=0">
```

```

<xsl:if test="count(institution)>0">
    <xsl:value-of select="institution"/>
    <text>, </text>
</xsl:if>
</xsl:if>
<xsl:if test="count(author)=0">
    <xsl:if test="count(organization)>0">
        <xsl:value-of select="organization"/>
        <text>, </text>
    </xsl:if>
</xsl:if>

```

Alle weiteren benötigten Elemente werden auf entsprechende Art und Weise für den Output definiert, d.h. sie erscheinen in der Form:

```

<xsl:if test=count(Element)>0>
    <xsl:value-of select="Elementwert"/>
    <text>Deskriptionszeichen</text>
</xsl:if>

```

Ich habe sämtliche Ausgabeaufforderung an eine Bedingung geknüpft, da in einem Literaturzitat immer auch Angaben fehlen können und in einem solchen Fall dann ein unnötiges Deskriptionszeichen im Output erscheinen würde.

### 5.3.2.2 Bücher als unselbständige Quelle

Werden Bücher als unselbständige Quelle benutzt, müssen außer den im vorherigen Kapitel aufgeführten Angaben noch die Angaben

- übergelagerter Titel
- und
- Seiten
- gemacht werden.

Die entsprechenden XML-Elemente sind

- <booktitle>
- und
- <pages>

Da eine derartige Literaturquelle nicht explizit durch das XML-Attribut *type* spezifiziert wird, bin ich davon ausgegangen, daß es sich genau dann um einen Beitrag aus einem

Buch handelt, wenn das Element `<booktitle>` im XML-Dokument vorhanden ist und somit durch folgende Anweisung aufgerufen wird:

```
<xsl:if test="count(booktitle)>0"> </xsl:if>
```

### 5.3.2.3 Zeitschriftenartikel

Zusätzlich zu den für alle Literaturtypen identischen Angaben muß folgendes aufgeführt werden:

- Titel der Zeitschrift oder Zeitung
- Bandzählung
- Heftnummer

oder

- Monat
- Seitenzählung

Die XML-Elemente dafür sind:

- `<journal>`
- `<volume>`
- `<number>`
- `<month>`
- `<pages>`

Hinsichtlich des Zeitschriftentitels muß unterschieden werden, ob im Ursprungsdokument das Attribut *alias* des Unterelements von `<journal>`, `<abbrev>`, vorhanden ist oder ob lediglich der Zeitschriftentitel durch `<journal>` angesprochen wird.

Mein Lösungsweg dafür ist nachstehender:

```
<xsl:choose>
  <xsl:when test="count(journal/abbrev/@alias)>0">
    <xsl:value-of select="journal/abbrev/@alias"/>.
  </xsl:when>
  <xsl:otherwise>
    <xsl:value-of select="journal"/>.
    <xsl:if test="starts-with(journal, .)">
      <xsl:value-of select="journal"/>
    </xsl:if>
  </xsl:otherwise>
</xsl:choose>
```

Auch für die Seitenzählung benötigte ich eine spezielle Lösung, da die Seiten in BIB-TEXML-Dokument angegeben werden wie im Beispiel

```
<pages>3--17</pages>.
```

Eine Darstellung mit zwei Gedankenstrichen ist jedoch nicht gewünscht, so daß ich auf die XPath-Ausdrücke `substring-after(string, marker)` und `substring-before(string, marker)` zurückgegriffen habe. Somit erhalte ich den Quelltext

```
<xsl:value-of select="substring-before(normalize-space(child::pages),'-')"/>
```

```
<xsl:value-of select="substring-after(normalize-space(child::pages),'-')"/>.
```

#### 5.3.2.4 WWW-Seite

Handelt es sich bei der zu zitierenden Quelle um eine WWW-Seite<sup>1</sup>, werden außer sämtlichen den Verfasser oder sonstigen Urhebern betreffenden Angaben, Titel und Jahr noch zwei weitere benötigt: die XML-Elemente

- `<school>` (optional)

und

- `<url>`

Das Element `<school>` bezieht sich auf eine Hochschule, von der eine WWW-Seite veröffentlicht wurde.

Da kein entsprechendes *type*-Attribut existiert, wird eine das Vorhandensein des Elements `<url>` abgefragt, um somit eine WWW-Seite als Quelle zu identifizieren.

#### 5.3.2.5 Darstellung von bibliographischen Angaben

DIN 1505 gibt genaue Regeln für Quellenangaben bezüglich der Reihenfolge der einzelnen Bestandteile und des Einsatzes von Deskriptionszeichen vor, für die Gestaltung z.B. durch Auszeichnungsschriften bietet sie lediglich Empfehlungen. Es sind jedoch auch andere Bibliographiestile möglich. Wichtig ist dabei nur, die gewählte Form auch durchgängig einzuhalten.

Um ein einheitliches Bibliographieformat zu erhalten, habe ich zwei XSLT-Stylesheets entwickelt, die die Möglichkeit zur Gestaltung von Literaturlisten geben sollen.

Ein Stylesheet, `bibdin.xsl`, soll eine an DIN 1505 angelehnte Darstellung ermöglichen. Ein weiteres Stylesheet, `bibalt.xsl` bietet eine Alternative zur Norm an.

---

<sup>1</sup> Diese Angaben stammen aus folgender Quelle:

Institut für Landesplanung und Raumforschung (1998): Handreichung für Studierende 1. Zitierregeln und Literaturverzeichnisse.[http://www.laum.uni-hannover.de/ilr/lehre/Handout/HR\\_Zitier5.htm](http://www.laum.uni-hannover.de/ilr/lehre/Handout/HR_Zitier5.htm). Datum des Zugriffs: 04.11.2001

Durch bibdin.xsl werden die Urheber eines Werks wie Nachnamen von Verfassern, Herausgeber, Institutionen oder Organisationen zur besseren Hervorhebung in Kapitälchen geschrieben, der Sachtitel wird kursiv dargestellt.

Im Stylesheet wird diese Auszeichnung folgendermaßen erreicht:

```
<font style="font-variant:small-caps">
```

bzw.

```
<i><xsl:value-of select="title"/></i>
```

Zur Illustration zeige ich nun die Gestalt der aus dem Stylesheet bibdin.xsl resultierenden Ausgabe für jeden Literaturtyp auf. Alternativ zur Angabe eines Verfassernamens kann jeweils ein Herausgeber mit nachgestelltem „(Hrsg.)“, eine Institution oder eine Organisation stehen. Mehrere Verfassernamen werden durch Semikolon voneinander getrennt. Bei mehr als drei Verfassern wird nur der erste ausgegeben, die übrigen werden durch „et al.“ ersetzt.

- Buch:

FAMILIENNAME Suffix, erster Vorname (ausgeschrieben) zweiter Vorname (abgekürzt): *Sachtitel*. Ausgabebezeichnung. Erscheinungsort: Verlag, Erscheinungsjahr

- Beitrag aus einem Buch:

FAMILIENNAME Suffix, erster Vorname (ausgeschrieben) zweiter Vorname (abgekürzt): *Sachtitel*. In: Übergelagerter Titel. Ausgabebezeichnung. Erscheinungsort: Verlag, Erscheinungsjahr. Seitenzählung „S.“

- Zeitungsartikel:

FAMILIENNAME Suffix, erster Vorname (ausgeschrieben) zweiter Vorname (abgekürzt): *Sachtitel*. In: Zeitschriftentitel. Bandzählung bzw. Heftnummer, (Erscheinungsjahr), Heftnummer „Nr.“, Seitenzählung „S.“

- WWW-Seiten:

FAMILIENNAME Suffix, erster Vorname (ausgeschrieben) zweiter Vorname (abgekürzt): *Sachtitel*. Erscheinungsjahr. Hochschule, Institution oder Organisation, „Stand“: URL

Nachfolgend soll eine beispielhafte Umwandlung nach diese Form angezeigt werden.

Das zugrunde liegende XML-Dokument sieht folgendermaßen aus:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<bibliography>
```

```
<bibitem type="article" label="ieee-ec:Back+Hammel+Schwefel:1997">
```

```
  <author>
```

```
    <firstname>Thomas</firstname>
```

```
    <lastname>Bäck</lastname>
```



```

</author>
<author>
  <firstname>Ulrich</firstname>
  <lastname>Hammel</lastname>
</author>
<author>
  <firstname>Hans-Paul</firstname>
  <lastname>Schwefel</lastname>
</author>
<title>Evolutionary Computation: Comments on the History and Current State</title>
<journal>
  <abbrev alias="IEEE-EC"/>
</journal>
<year>1997</year>
<type>Paper</type>
<volume>1</volume>
<number>1</number>
<month>
  <abbrev alias="apr"/>
</month>
<pages>3--17</pages>
</bibitem>
</bibliography>

```

Die Anzeige im Browser ist unten angeführt:

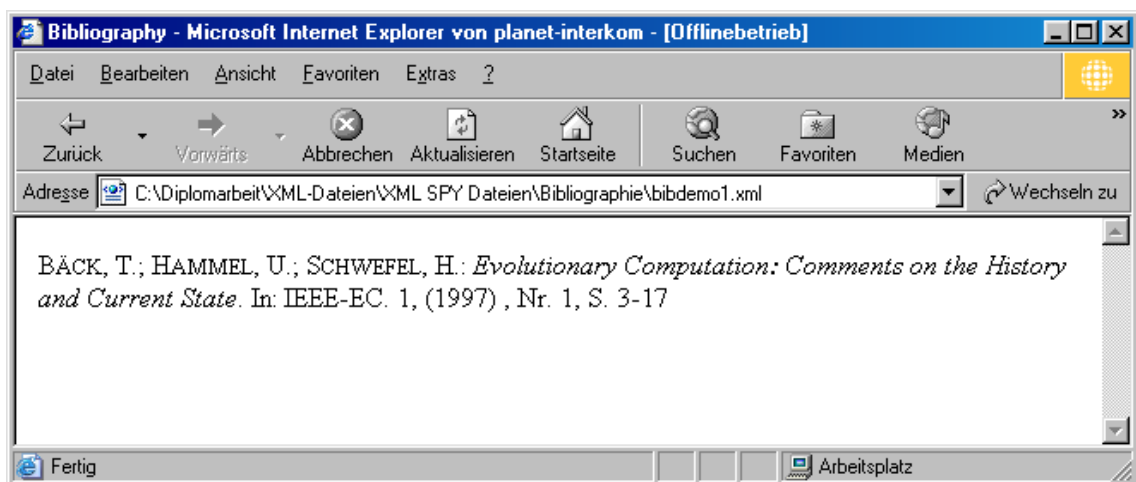


Abbildung 2: An DIN 1505 angelehnte Form von Literaturangaben

Im alternativen Stylesheet bibalt.xsl wird teilweise eine andere Reihenfolge verwendet, Urheberangaben sind fett geschrieben und auch die Deskriptionszeichen unterscheiden sich von der ersten Ausgabeform.

Der erste Vorname eines Verfassers ist hier abgekürzt. Mehrere Verfasser werden durch Komma voneinander getrennt, vor dem letzten Verfassernachname steht ein „und“, z.B.

Familienname1, Vorname1, Familienname2, Vorname2 und Familienname3, Vorname3.

Sind mehr als drei Verfasser angegeben, wird, entsprechend der ersten Stylesheet-Variante, nur der erste ausgegeben, es folgt ein „u.a.“.

Auch hier stelle ich im Anschluß die Form einer derartigen Zitiermöglichkeit dar.

- Buch

**Familienname Suffix, erster Vorname** (abgekürzt) **zweiter Vorname** (abgekürzt) (Erscheinungsjahr): Sachtitel. Ausgabebezeichnung. Verlag, Erscheinungsort.

- Beitrag aus einem Buch:

**Familienname Suffix, erster Vorname** (abgekürzt) **zweiter Vorname** (abgekürzt) (Erscheinungsjahr): Sachtitel. In: Herausgeber (Hrsg.): Übergelagerter Titel. Ausgabebezeichnung. Verlag, Erscheinungsort. Seitenzählung „S.“

- Zeitungsartikel:

**Familienname Suffix, erster Vorname** (abgekürzt) **zweiter Vorname** (abgekürzt) (Erscheinungsjahr): Sachtitel. Zeitschriftentitel. Erscheinungsort. Bandzählung, Heftnummer „Nr.“, Seitenzählung „S.“

- WWW-Seiten:

**Familienname Suffix, erster Vorname** (abgekürzt) **zweiter Vorname** (abgekürzt) (Erscheinungsjahr): Sachtitel. Hochschule, Institution oder Organisation, URL. „Datum des Zugriffs:“

Die oben angeführte XML-Datei soll auch für die alternative Darstellungsweise als Beispiel dienen. Im Browser präsentiert sich eine solche wie folgt:

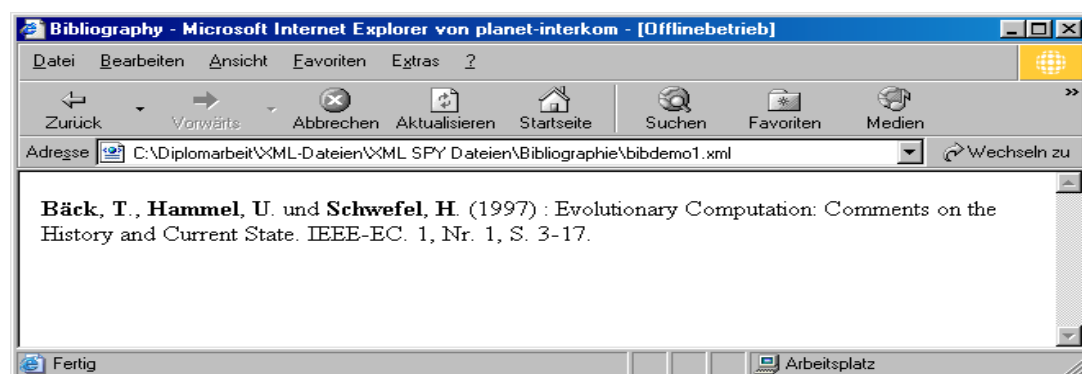


Abbildung 3: Alternative Darstellungsweise

### 5.3.2.6 Sortieren der bibliographischen Angaben

Oft ist es hilfreich, Daten nach bestimmten Kriterien zu sortieren. So können große Datenmengen übersichtlich geordnet werden.

Das gilt auch für Literaturverzeichnisse. Bibliographien werden einer wissenschaftlichen Arbeit normalerweise alphabetisch geordnet angefügt, so daß es sinnvoll wäre, Literaturverzeichnisse nach Alphabet sortieren zu können.

Mit den Funktionen von XSLT habe ich drei Stylesheets entworfen, die eine Sortierung nach verschiedenen Kriterien erlauben.

1. Das Stylesheet Sort\_author.xls sortiert Verfasseramen alphabetisch ein. Dabei werden erst die Nachnamen berücksichtigt, anschließend die Vornamen. Diese Sortierung wird erreicht durch folgende XSLT-Anweisung:

```
<xsl:sort select="author/lastname"/>
```

```
<xsl:sort select="author/firstname"/>
```

2. Das Stylesheet Sort\_title.xls ermöglicht es, eine große Bibliographiedatei alphabetisch nach Sachtiteln zu sortieren. Die entsprechende XSLT-Anweisung ist

```
<xsl:sort select="title"/>.
```

3. Eine dritte Sortiervariante erfüllt das Stylesheet Sort\_year das es erlaubt, Bibliographie-Daten nach dem Erscheinungsjahr zu sortieren. Dabei wird das früheste Erscheinungsjahr zuerst angegeben, dann wird aufsteigend sortiert. Als zweites Sortierkriterium wird der Verfasser berücksichtigt, d.h. gibt es mehrere Literaturangaben mit dem selben Erscheinungsjahr, wird innerhalb dessen alphabetisch nach Verfasseramen sortiert. Unten stehender XSLT-Befehl führt die gewünschte Funktion aus.

```
<xsl:sort order="ascending" select="year"/>
```

```
<xsl:sort select="author/lastname"/>
```

Folgende Abbildung zeigt eine Sortierung nach Verfasseramen.

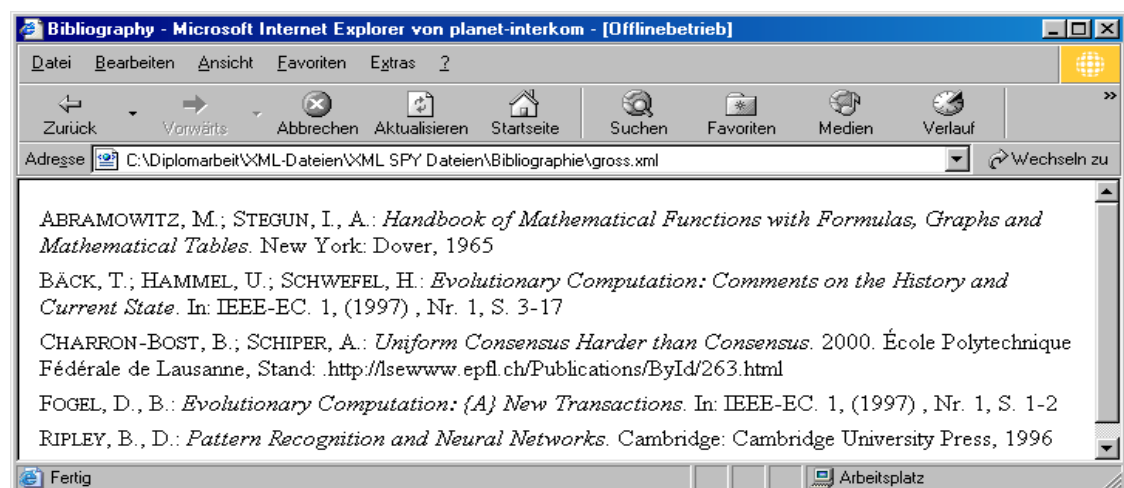


Abbildung 4: Sortierung nach Verfasseramen

### 5.3.2.7 Auswahl einzelner Literaturangaben

Besonders im Hinblick auf die Verwendung Literaturzusammenstellungen im Internet schien es mir sinnvoll zu sein, die Auswahl einzelner Literaturangaben aus einer großen Bibliographiedatei zu realisieren.

Als praktisches Beispiel für den Einsatz einer solchen Funktion kann z.B. der Internet-auftritt einer Hochschule genannt werden. So wäre es denkbar, aus einer Datei, in der sämtliche Veröffentlichungen von Professoren der Hochschule gespeichert sind, genau die Quellenangaben herauszufiltern, in denen ein bestimmter Professor als Autor genannt wird. Eine solche Auswahl könnte dann beispielsweise in der Homepage eines Professors verwendet werden, der explizit die Literatur aufführen will, in der er als Autor genannt ist.

Programmiertechnisch umsetzbar wird diese Voraussetzung, indem allen übrigen Anweisungen im Stylesheet die Bedingung vorangestellt wird, daß ein Output nur dann erfolgen soll, wenn das Element `<author/lastname>` einen bestimmten Wert annimmt. Um nachfolgendes Beispiel zu veranschaulichen, habe ich diesen Wert auf „Riekert“ gesetzt. Somit werden alle Werke ausgegeben, in denen Herr Riekert als Autor angegeben ist.

```
<xsl:if test="author/lastname[.='Riekert']"></xsl:if>
```

### 5.3.2.8 Generierung eines Winword-Ausgabeformats

Eine weitere und äußerst wichtige praktische Einsatzmöglichkeit von Literaturverzeichnissen ist die Verwendung in einer wissenschaftlichen Arbeit. Da Bibliographien im Moment noch mühsam von Hand erstellt werden müssen, sah ich es als zusätzliches Ziel meiner Arbeit an, Quellenangabe auf XML-Basis in ein Format zu bringen, das problemlos in Winword eingefügt werden kann. Wie bereits erwähnt, habe ich mich auf Winword konzentriert, das es wohl das am häufigsten verwendete Textverarbeitungsprogramm ist.

Bewußt habe ich mich deshalb zu Beginn meiner Entwicklungen für einen HTML-Output entschieden, da Daten im HTML-Format leicht in Winword übernommen werden können.

Als Schriftart für die Ausgabedatei habe ich Times New Roman ausgewählt, da diese Schriftart sehr häufig für wissenschaftliche Arbeiten verwendet wird. Problemlos kann aber auch Arial eingesetzt werden

Der eingesetzte Text erscheint in der Schriftgröße 10. Das entspricht im HTML-Code der Schriftgröße 3. Wird hier die Schriftgröße 4 verwendet, wird der Text im Winword-Dokument mit dem Schriftgrad 14 ausgegeben.

Eine Bibliographie besteht üblicherweise aus vielen verschiedenen Quellenangaben, zwischen denen zur besseren Strukturierung eine Leerzeile steht.

Um Literaturdaten im HTML-Format in dieser Form in einem Word-Dokument einfügen zu können, muß jede einzelne Literaturangabe in einer separaten Tabelle stehen.

Im Laufe meiner Entwicklungen habe ich zunächst jedes Zitat als Absatz formatiert. Dazu habe ich für jedes <bibitem> -Element einen Output zwischen den Tags <p> und </p> vorgesehen. Beim Testen in Winword mußte ich jedoch feststellen, daß die Absätze nicht übernommen wurden. Durch die Verwendung von blinden Tabellen dagegen wird die Bibliographie wie gewünscht formatiert.

Eingefügt werden kann eine Literaturdatei im HTML-Format über das Menü durch Auswählen des Menüpunkts „Einfügen“ – „Datei“, wie auch unten stehende Abbildung zeigt. Anschließend kann die einzufügende Datei ausgewählt werden.

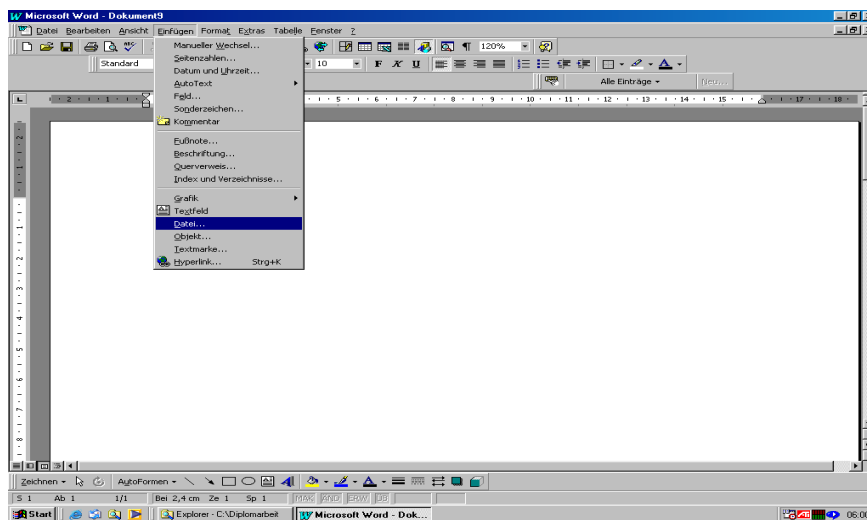


Abbildung 5: Einfügen einer HTML-Datei in Winword

### 5.3.3 Ergebnis

Ziel meiner Arbeit war, Literaturangaben für wissenschaftliche Arbeiten wie z.B. für Diplomarbeiten in ein einheitliches Format zu bringen und deren Erstellung dadurch zu erleichtern.

Dazu habe ich zunächst Überlegungen hinsichtlich des optimalen Ausgabeformats angestellt und mich dann für das HTML-Format entschieden.

Danach hab ich die verschiedenen Literaturquellen unterschieden, ihre Eigenschaften und Bestandteile untersucht und versucht, diese durch XSLT-Anweisungen wiederzugeben.

Um Literaturverzeichnissen ein einheitliches Format zu geben, habe ich zwei XSLT-Stylesheets entwickelt. Durch diese Programme wird es möglich, den einzelnen Bestandteilen einer Literaturangabe eine feste, vordefinierte Form zuzuweisen, um eine komplette Literatursammlung dadurch zu vereinheitlichen. So ist es nicht mehr möglich, unterschiedliche Formatierungen oder Deskriptionszeichen innerhalb einer Bibliographie zu verwenden.

Eines der beiden Stylesheets gibt die Möglichkeit einer Ausgabe, die in Bezug auf die Reihenfolge der einzelnen Bestandteile und der Verwendung von Deskriptionszeichen stark an DIN 1505 angelehnt ist. Das zweite Stylesheet bietet eine Alternative und unterscheidet sich vor allem hinsichtlich der Textauszeichnung und der zur Trennung verwendeten Zeichen, geringfügig aber auch in der Reihenfolge der Elemente einer Literaturangabe.

Bibliographien können mit Hilfe von XSLT-Stylesheets nach verschiedenen Kriterien sortiert werden. Aktuell sind für diese Funktion drei separate Dateien mit einer jeweils festen Sortiervorgabe vorgesehen.

Ein solcher Prototyp bietet Voraussetzungen für den praktischen Einsatz: so gelingt es z.B., Literatursammlungen alphabetisch zu sortieren, so wie sie üblicherweise in wissenschaftlichen Arbeiten erscheinen. Außerdem kann nach Sachtitel und Erscheinungsjahr sortiert werden.

Das Stylesheet für die Auswahl einzelner Literaturangaben aus einer großen Literaturdatendatei ist so konzipiert, daß definitiv ein bestimmter Name als festes Auswahlkriterium angegeben ist. Nach der aktuellen Struktur des Programms werden alle Quellenangaben ausgegeben, in denen der angegebene Name als Nachname eines Verfassers auftaucht. Eine ins HTML-Format transformierte BIBTEXML-Datei kann in Winword eingefügt werden und somit konkret bei der Erstellung von Literaturverzeichnissen in wissenschaftlichen Arbeiten unterstützen.

## 6 Abschließende Zusammenfassung und Ausblick

Im Rahmen meiner Arbeit habe ich einen Prototyp entwickelt, mit dem ein Ansatz für eine Unterstützung bei der Literaturverzeichniserstellung geboten werden könnte. Dafür habe ich mich auf die Möglichkeiten der Meta-Auszeichnungssprache XML, ihrer Stylesheetsprache XSL und des darin enthaltenen Transformationsprogramms XSLT konzentriert.

Anlaß zu vorliegender Arbeit gab der Umstand, daß an Dokumente angehängte Literatursammlungen zumeist von Hand erstellt werden müssen und dadurch eine oft uneinheitliche Form erhalten, da selbst innerhalb einer Datei unterschiedliche Textauszeichnungen oder Anordnungen der einzelnen Bestandteile verwendet werden.

Aufsetzend auf die bereits bestehende Möglichkeit, Literaturdaten aus dem Internet im BIBTEX-Format in XML umwandeln zu können und dadurch eine klare Datenstruktur zu gewährleisten, habe ich die Möglichkeiten des Transformationsprogramms XSLT eingesetzt, um diese Daten weiterverarbeiten zu können.

So sind prototyphaft diverse XSLT-Stylesheets entstanden, die die gewünschten Funktionalitäten hinsichtlich der Vereinheitlichung von Bibliographien und deren Darstellung bieten sollen. Dabei werden verschiedene Darstellungsformen zur Verfügung gestellt. Des weiteren besteht die Möglichkeit, ganze Literaturverzeichnisdateien zu sortieren und eine bestimmte Angabe auszuwählen. Die durch XSLT generierte HTML-Datei kann leicht in das Textverarbeitungsprogramm Winword eingefügt werden, dazu eignet sich das HTML-Format natürlich auch für die Verwendung der Daten in WWW-Seiten.

Für zukünftige Entwicklungen könnten die von mir entwickelten XSLT-Funktionalitäten den Ausgangspunkt für eine Formularschnittstelle bieten um so die Möglichkeit eines Online-Einsatzes zu realisieren. Wie ein solches Formular aussehen könnte, zeigt nachstehende Abbildung.

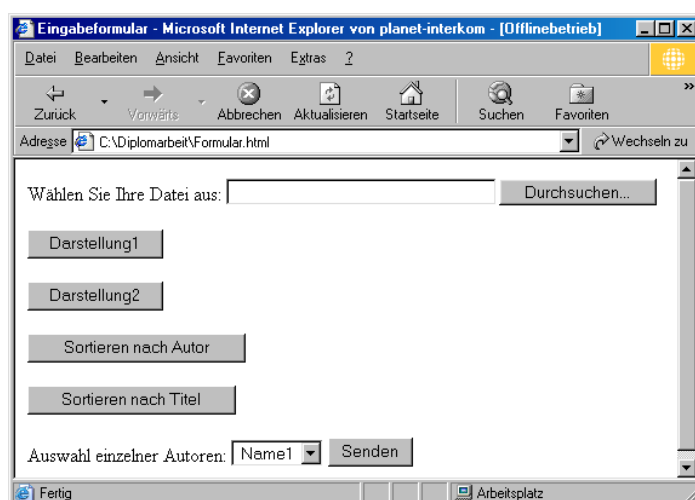


Abbildung 6: Online-Formular

Die benötigten technischen Funktionen ließen sich mit PHP umsetzen. Mit PHP können Formulare leicht ausgewertet werden. Außerdem besitzt PHP einen integrierten Prozessor, der vom Programm aus gestartet werden kann. Eine solche Methode der Auswertung von Formularen wird im Moment im Rahmen einer Diplomarbeit an der Hochschule der Medien (HdM) realisiert.

Sortierkriterien für Literaturangaben wie sie in Kapitel 5.3.2.6 dargestellt sind könnten noch allgemeiner formuliert werden, so daß ein einziges Stylesheet genügen müßte.

Ein weiterer Schritt wäre, mit Java Zusatzfunktionen zu programmieren, wie z.B. das Weglassen einer bestimmten Literaturangabe im Ausgabedokument.



## Anhang A: Programmbeispiele

Nachfolgend sind Auszüge des Programmcodes aufgeführt, der für die an DIN 1505 angelehnte Ausgabeform entwickelt wurde.

### A.1 Ausgabe eines Autors

```
<!--Folgende Ausgabe gilt für einen vorhandenen Autor-->
<xsl:if test="count(author)=1">
  <font style="font-variant:small-caps">
    <xsl:value-of select="author/lastname"/>
  </font>
  <xsl:if test="count(suffix)>0">
    <xsl:value-of select="suffix"/>
  </xsl:if>
  <text>, </text>
  <!-- Der firstname soll abgekürzt ausgegeben werden.-->
  <xsl:if test="count(author/middlename)=0">
    <xsl:value-of select="normalize-space(substring(author/firstname,1,1))"/>
    <text>.</text>
    <text>: </text>
  </xsl:if>
  <xsl:if test="count(author/middlename)>0">
    <xsl:value-of select="normalize-space(substring(author/firstname,1,1))"/>
    <text>.</text>
    <text>, </text>
  </xsl:if>
  <xsl:if test="count(author/middlename)>0">
    <xsl:value-of select="author/middlename"/>
    <text>: </text>
  </xsl:if>
</xsl:if>
```

### A.2 Ausgabe für Bücher und Beiträge aus Büchern

```
<xsl:if test="@type[.='book']">
  <xsl:if test="count(booktitle)>0">
    <text>In: </text>
```

```

        <xsl:value-of select="booktitle"/>
        <text>. </text>
    </xsl:if>
    <xsl:if test="count(edition)>0">
        <xsl:value-of select="edition"/>
        <text>. </text>
    </xsl:if>
    <xsl:if test="count(address)>0">
        <xsl:value-of select="address"/>
        <text>: </text>
    </xsl:if>
    <xsl:if test="count(publisher)>0">
        <xsl:value-of select="publisher"/>
        <text>, </text>
    </xsl:if>
    <xsl:if test="count(year)>0">
        <xsl:value-of select="year"/>
    </xsl:if>
    <xsl:if test="count(booktitle)>0">
    <xsl:if test="count(pages)>0">
        <xsl:value-of select="pages"/>
        <text>. S. </text>
    </xsl:if>
    </xsl:if>
</xsl:if>

```

### A.3 Ausgabe für Zeitungsaufsätze

```

<xsl:if test="@type[.='article']">

```

<!--Der Titel der Zeitschrift wird angegeben. Dabei muß unterschieden werden, ob das Attribut "alias" des Unterelements "abbrev" vorhanden ist oder ob lediglich der Zeitschriftentitel angesprochen wird-->

```

    <text>In: </text>
    <xsl:choose>
        <xsl:when test="count(journal/abbrev/@alias)>0">
            <xsl:value-of select="journal/abbrev/@alias"/>
            <text>. </text>
        </xsl:when>
        <xsl:otherwise>

```

```

        <xsl:value-of select="journal"/>.
        <xsl:if test="starts-with(journal, .)">
            <xsl:value-of select="journal"/>
            <text>. </text>
        </xsl:if>
    </xsl:otherwise>
</xsl:choose>
<xsl:if test="count(volume)>0">
    <xsl:value-of select="volume"/>
    <text>, </text>
</xsl:if>
<xsl:if test="count(year)>0">
    (<xsl:value-of select="year"/>)
    <text>, </text>
</xsl:if>
<xsl:if test="count(number)>0">
    <text>Nr. </text>
    <xsl:value-of select="number"/>
    <text>, </text>
</xsl:if>
<xsl:if test="count(number)=0">
<xsl:if test="count(month/abbrev/@alias)>0">
    <xsl:value-of select="month/abbrev/@alias"/>
    <text>, </text>
</xsl:if>
</xsl:if>
<!--Zum Schluß werden die Seiten angegeben. Hierbei soll das Format aus dem ursprünglichen
xml-Dokument (Seitenzahl 2x - Seitenzahl) geändert werden in Seitenzahl - Seitenzahl-->
<xsl:if test="count(pages)>0">
    <text>S. </text>
    <xsl:value-of select="substring-before(normalize-space(child::pages),'-')"/>
    <xsl:value-of select="substring-after(normalize-space(child::pages),'-')"/>
</xsl:if>
</xsl:if>

```

## Anhang B: Elemente einer BIBTEXML-Datei

Nachfolgend sind die Unterelemente von `<bibitem>` aufgelistet<sup>1</sup>. Nicht alle dieser Felder werden tatsächlich bei der Erstellung eines Literaturverzeichnisses im von mir gewünschten Sinne benötigt.

`<address>`

`<annotate>`

`<author>`

`<booktitle>`

`<chapter>`

`<crossref>`

`<edition>`

`<editor>`

`<howpublished>`

`<institution>`

`<journal>`

`<key>`

`<keywords>`

`<month>`

`<note>`

`<number>`

`<organization>`

`<pages>`

`<publisher>`

`<school>`

`<series>`

`<title>`

`<type>`

`<volume>`

---

<sup>1</sup> Diese Auflistung geschieht in starker Anlehnung an die Diplomarbeit zu diesem Thema, die an der Eidgenössischen Technischen Hochschule (ETH) Zürich erstellt wurde.

<year>

<notstandard>

<URL>

<ISSN>

<ISBN>

<abstract>

<contents>

## Literaturverzeichnis

**World Wide Web Consortium (W3C)** (1999): XSL Transformations (XSLT), Version 1.0. W3C Recommendation 16. November 1999.

<http://www.w3.org/TR/1999/REC-xslt-19991116>

Datum des Zugriffs: 04.11.2001

**Lurati, B. und Previtali, L.** (2000/2001): BIBTEXML. Eidgenössische Technische Hochschule, Zürich. Diplomarbeit.

[http://bibtexml.org/documents/da\\_2001\\_02.pdf](http://bibtexml.org/documents/da_2001_02.pdf).

Datum des Zugriffs: 04.11.2001

**Niedermair, E. und Niedermair, M.** (2001): Das große Buch XML. Data Becker, Paderborn.

**Behme, H. und Mintert, S.** (2000): XML in der Praxis. Professionelles Web-Publishing mit der Extensible Markup Language. 2., erw. Aufl. Addison-Wesley, München.

**Goosens, M., Mittelbach, F. und Samarin, A.** (1994): Der LATEX-Begleiter. Addison-Wesley, München.

**Deutsches Institut für Normung E.V. (DIN)** (1984): DIN 1505, Teil 2. Titelangaben von Dokumenten: Zitierregeln. Beuth Verlag GmbH, Berlin.

**Institut für Landesplanung und Raumforschung** (1998): Handreichung für Studierende 1. Zitierregeln und Literaturverzeichnisse.

[http://www.laum.uni-hannover.de/ilr/lehre/Handout/HR\\_Zitier5.htm](http://www.laum.uni-hannover.de/ilr/lehre/Handout/HR_Zitier5.htm).

Datum des Zugriffs: 04.11.2001

## Erklärung

Hiermit erkläre ich, daß ich die vorliegende Diplomarbeit selbständig angefertigt habe. Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt. Wörtlich oder sinngemäß übernommenes Gedankengut habe ich als solches kenntlich gemacht.

---

Ort, Datum

---

Unterschrift